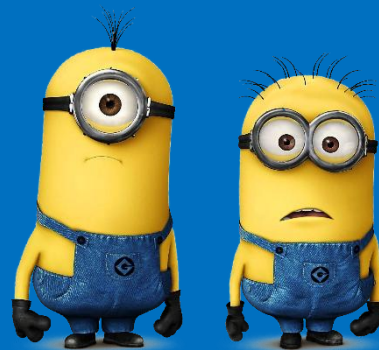




Programming
Languages

Лабораторная работа #3

Базовый синтаксис языка Python: функции. Линейные алгоритмы



ЛАБОРАТОРНАЯ РАБОТА # 3

Базовый синтаксис языка Python: функции. Линейные алгоритмы

Цель работы

Ознакомится с принципами хранения данных в компьютере; освоить базовый синтаксис языка Python; закрепить создания интерактивных программ с использованием линейных алгоритмов (последовательного выполнения кода) и математической библиотеки функций `math`.

Основное задание

Разработать интерактивную программу «What is My Age in Seconds» («Каков мой возраст в секундах»), которая принимает дату рождения пользователя, высчитывает количество прожитых пользователем секунд и выводит результат на экран монитора.

Индивидуальное задание

Необходимо запрограммировать на языке Python вычисление соответствующей функции, которая была дана на лабораторной работе по теме **«Линейные алгоритмы»** по параллельной специализированной дисциплине «Основы алгоритмизации» (соответствующие функции можно взять из модуля **`math`**).

Дополнительное задание*

Согласно официальному сайту Европейского Центрального Банка (источник: <http://www.ecb.europa.eu>) в настоящий момент наши соседи используют в обращении **7** номиналов банкнотов евро - 5, 10, 20, 50, 100, 200 и 500 евро. Банкноты выполнены в едином стиле, но в зависимости от номинала различаются цветовой гаммой и размерами.

На лицевой стороне банкнотов изображены окна и ворота, являющиеся символом открытости и европейского духа сотрудничества. На оборотной стороне банкнотов изображены мосты символизирующие открытое общение.

Монетный ряд включает **8** монет достоинством 1, 2, 5, 10, 20 и 50 евроцентов (1 евро = 100 евроцентов), а также 1 и 2 евро.

На лицевой стороне (аверс) каждой из монет отчеканен рисунок, отображающий те или иные черты того или иного государства.

На оборотной стороне (реверс) монет отчеканен один из трех общих для всех стран рисунков с изображением контуров Европы, окруженной 12 звездами Европейского Союза. Общий рисунок изображает карту ЕС на фоне параллельных линий, протянувшихся к 12 звездам на флаге ЕС. На монетах 1, 2 и 5 центов изображено место Европы на карте мира, а монеты достоинством 10, 20 и 50 центов показывают Европу как группу отдельных государств. На монетах в 1 и 2 евро дается изображение карты Европы без границ между отдельными государствами.



- 1) Необходимо написать программу для специального банкомата (АТМ). Банкомат должен выдавать любую ведённую сумму пользователем, но оптимальным способом. Вначале выдаются крупные банкноты, потом меньше и так вплоть до одного евроцента. К примеру, пользователь запрашивает сумму в 587 евро и 99 центов. Банкомат ему должен выдать следующее: банкноты номиналом 500, 50, 20, 10 и 5 евро, а затем монеты: 50, 20x2, 5, 2x2 и 1 евроцент.
- 2) Необходимо написать приложение для разменного автомата. Работа автомат заключается в следующем: автомат принимает у пользователя купюру и выдаёт ему купюры меньшего номинала. То же самое происходит и с монетами.



Студенты, которые находятся под нечётным номером в списках групп, берут себе первый вариант задания, а все остальные – второй.

Если кому не нравится работать с евро (- это может звучать странно ☺), могут взять любую другую валюту, которая больше нравится, никаких ограничений.

Требования к выполнению

- 1) Программа должна обязательно быть снабжена комментариями на английском языке, в которых необходимо указать краткое предназначение программы, номер лабораторной работы и её название, версию программы, ФИО разработчика, номер группы и дату разработки.
- 2) Каждая программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом (да-да, пусть даже пока в консольном варианте).

Перед началом работы всегда думайте о нуждах конечных пользователей, только так можно создать по-настоящему качественный продукт!



Best of LUCK with it, and remember to HAVE FUN while you're learning :)

Victor Ivanchenko



Контрольные вопросы

Список математических функций модуля `math`

Теоретико-численные функции и функции представления

Name of function	Description
<code>math.ceil(x)</code>	Возвращает округленное x как ближайшее целое значение типа <code>float</code> , большее или равное x (округление "вверх").
<code>math.copysign(x, y)</code>	Возвращает число x со знаком числа y . На платформе, поддерживающей знак нуля <code>copysign(1.0, -0.0)</code> даст <code>-1.0</code> .
<code>math.fabs(x)</code>	Возвращает абсолютное значение (модуль) числа x . В Python есть встроенная функция <code>abs</code> , но она возвращает модуль числа с тем же типом, что число, здесь же всегда <code>float abs (fabs)</code> .
<code>math.factorial(x)</code>	Возвращает факториал целого числа x , если x не целое возбуждается ошибка <code>ValueError</code> .
<code>math.floor(x)</code>	В противоположность <code>ceil(x)</code> возвращает округленное x как ближайшее целое значение типа <code>float</code> , меньшее или равное x (округление "вниз").
<code>math.fmod(x, y)</code>	Аналогична функции <code>fmod(x, y)</code> библиотеки C. Отметим, что это не то же самое, что выражение Python <code>x%y</code> . Желательно использовать при работе с объектами <code>float</code> , в то время как <code>x % y</code> больше подходит для <code>int</code> .
<code>math.frexp(x)</code>	Представляет число в экспоненциальной записи $x = m * 2^e$ и возвращает мантиссу m (действительное число, модуль которого лежит в интервале от 0.5 до 1) и порядок e (целое число) как пару чисел (m, e) . Если $x=0$, то возвращает $(0.0, 0)$
<code>math.fsum(iterable)</code>	Возвращает <code>float</code> сумму от числовых элементов итерируемого объекта.
<code>math.isinf(x)</code>	Проверяет, является ли <code>float</code> объект x плюс или минус бесконечностью, результат соответственно <code>True</code> или <code>False</code> .
<code>math.isnan(x)</code>	Проверяет, является ли <code>float</code> объект x объектом <code>NaN</code> (not a number).
<code>math.ldexp(x, i)</code>	Возвращает значение $x * 2^i$, то есть осуществляет действие, обратное функции <code>math.frexp(x)</code> .
<code>math.modf(x)</code>	Возвращает часть, идущую после запятой и целую часть от <code>float</code> числа. Оба результата сохраняют знак исходного числа x и представлены типом <code>float</code>
<code>math.trunc(x)</code>	Возвращает целую часть числа x в виде <code>int</code> объекта